# Worst–Case Analysis of Weber's Algorithm

Christian Lavault, Sidi Mohamed Sedjelmaci

# Worst–Case Analysis of Weber's GCD Algorithm

Christian Lavault* and S. Mohamed Sedjelmaci
LIPN, CNRS UPRES-A 7030
Université Paris 13, F-93430 Villetaneuse

**Abstract**

Recently, Ken Weber introduced an algorithm for finding the $(a, b)$-pairs satisfying $au + bv \equiv 0 \pmod{k}$, with $0 < |a|, |b| < \sqrt{k}$, where $(u, k)$ and $(v, k)$ are coprime. It is based on Sorenson's and Jebelean's "$k$-ary reduction" algorithms. We provide a formula for $N(k)$, the maximal number of iterations in the loop of Weber's GCD algorithm.

**Keywords:** Integer greatest common divisor (GCD); Complexity analysis; Number theory.

## 1 Introduction

The greatest common divisor (GCD) of integers $a$ and $b$, denoted by $\gcd(a, b)$, is the largest integer that divides both $a$ and $b$.

Recently, Sorenson proposed the "right-shift $k$-ary algorithm" [7]. It is based on the following reduction. Given two positive integers $u > v$ relatively prime to $k$ (i.e., $(u, k)$ and $(v, k)$ are coprime), two integers $a$, $b$ can be found that satisfy

$$au + bv \;\equiv\; \pmod{k} \quad \text{with} \quad 0 < |a|,\ |b| < \sqrt{k}. \tag{1}$$

If we perform the transformation $(u, v) \longmapsto (u', v')$ (also called "$k$-ary reduction"), where $(u', v') = \Big(|au + bv|/k, \min(u, v)\Big)$, which replaces $u$ with $u' = |au + bv|/k$, the size of $u$ is reduced by roughly $1/2 \ \log_2(k)$ bits. Sorensen suggests table lookup to find sufficiently small $a$ and $b$ satisfying (1). By contrast, Jebelean [2, 3] and Weber [8] both propose an easy algorithm, which finds such small $a$ and $b$ that satisfy (1) with time complexity $O(n^2)$, where $n$ represents the number of bits in

*E-mail: `Christian.Lavault@lipn.univ-paris13.fr`

the two inputs. This latter algorithm we call the "Jebelean-Weber algorithm", or *JWA* for short.

The present work focuses on the study of $N(k)$, the maximal number of iterations of the loop in *JWA*, in terms of $t = t(k, c)$ as a function of two coprime positive integers $c$ and $k$ $(0 < c < k)$. Notice that this exact worst-case analysis of the loop does not provide the greatest lower bound on the complexity of *JWA*: it does not result in the optimality of the algorithm.

In the next Section 2, an upper bound on $N(k)$ is given, in Section 3, we show how to find explicit values of $N(k)$ for every integer $k > 0$. Section 4 is devoted to the determination of all integers $c > 0$, which achieve the maximal value of $t(k, c)$ for every given $k > 0$; that is the worst-case ocurrences of *JWA*. Section 5 contains concluding remarks.

## 2 An Upper Bound on $N(k)$

Let us recall the *JWA* as stated in [5, 8]. The first instruction $c := x/y \bmod k$ in *JWA* is not standard. It means that the algorithm finds $c \in [1, k-1]$, such that $cy = x + nk$, for some $n$ (where $x, y, k, c$, and $n$ are all integers).

---

Input: $x, y > 0$, $k > 1$, and
$\gcd(k, x) = \gcd(k, y) = 1$.
Output: $(n, d)$ such that
$0 < n,\ |d| < \sqrt{k}$, and $ny \equiv dx \pmod{k}$.
  $c := x/y \bmod k$ ;
  $f_1 = (n', d') := (k, 0)$ ;
  $f_2 = (n'', d'') := (c, 1)$ ;
**while** $n'' \geq \sqrt{k}$ **do**
  $f_1 := f_1 - \lfloor n'/n'' \rfloor f_2$ ;
  **swap** $(f_1, f_2)$
**endwhile**
**return** $f_2$

---

Notice that the loop invariant is $n' |d''| + n'' |d'| = k$. When $(n, d)$ is the output result of *JWA*, the pairs $(a, b) = (d, -n)$ and $(-d, n)$ meet property (1).

## 2.1 Notation

In *JWA*, the input data are the positive integers $k$, $u$ and $v$. However, for the purpose of the worst-case complexity analysis, we consider $c = u/v \bmod k$ in place of the pair $(u, v)$. Therefore, the actual input data of *JWA* are regarded as being $k$ and $c$, such that $0 < c < k$, and $\gcd(k, c) = 1$.

Throughout, we use the following notation. The sequence $(n_i, d_i)$ denotes the successive pairs produced by *JWA* when $k$ and $c$ are the input data. Let $t = t(k, c)$ denote the number of iterations of the loop of *JWA*; $t$ must satisfy the following inequalities:

$$n_t < \sqrt{k} < n_{t-1} \qquad \text{and} \qquad 0 < n_t,\ |d_t| < \sqrt{k}, \tag{2}$$

where finite sequence $D = (d_i)$ is defined recursively for $i = -1,\ 0\ ,\ 1, \ldots,\ (t-2)$ as

$$\begin{aligned} d_{i+2} = &\ d_i \ - \ q_{i+2}\, d_i \quad \text{with} \quad d_{-1} = 0 \ \text{ and } \ d_0 = 1 \\ q_{i+2} = &\ \lfloor n_i / n_{i+1} \rfloor \qquad \text{with} \quad n_{-1} = k \ \text{ and } \ n_0 = c. \end{aligned} \tag{3}$$

We denote by $Q = (q_i)$ the finite sequence of partial quotients defined in (3). The sequence $D$ is uniquely determined from the choice of $Q$ (i.e., $D = D(Q)$), since the initial data $d_{-1}$ and $d_0$ are fixed and $D$ is an increasing function of the $q_i$'s in $Q$. Let $(F_n)$ $(n = 0, 1, \ldots)$ be the Fibonacci sequence, we define $m(k)$ by

$$m(k) = \ \max\Big\{ i \geq 0 \mid F_{i+1} \leq \sqrt{k} \Big\} \qquad \text{with} \quad i \in \mathbb{N}.$$

For every given integer $k > 0$, the maximal number of iterations of the loop of *JWA* is:

$$N(k) = \ \max\Big\{ t(k, c) \mid 0 < c < k \ \text{ and } \ \gcd(k, c) = 1 \Big\}.$$

## 2.2 Bounding $N(k)$

**Lemma 2.1.** *With the above notation,*

*(i)* $|d_t| \geq F_{t+1}$.

*(ii)* $N(k) \leq m(k)$.

*Proof.*
*(i)* The proof is by induction on $t$.

- *Basis:* $|d_{-1}| = 0 = F_0$, $|d_0| = 1 = F_1$, and $|d_1| = q_1 \geq 1 = F_2$.
- *Induction step:* For every $i \geq 0$, suppose $|d_j| \geq F_{j+1}$ for $j = -1,\ 0,\ 1, \ldots,$ $(i-1)$. Then,

$$|d_i| \ = \ |d_{i-2}| \ + \ q_i|d_{i-1}| \ \geq \ |d_{i-2}| \ + \ |d_{i-1}| \ \geq \ F_{i-1} + F_i \ = \ F_{i+1}$$

and *(i)* holds.

*(ii)* $F_{t+1} \leq |d_t| < \sqrt{k}$. Hence $t = t(c,k) \ \leq \ m(k)$, and also $N(k) \leq m(k)$. $\square$

Note that the following inequalities also hold

$$\phi^{m-1} \ < \ F_{m+1} \ le \ \sqrt{k} \ < \ F_{m+2} \ < \ \phi^{m+1},$$

where $\phi = (1+\sqrt{5})/2$ is the golden ratio.

From Lemma 2.1 and the above inequalities, an explicit expression of $m(k)$ is easily derived,

$$m(k) = \lfloor \log_\phi(\sqrt{k}) \rfloor \quad \text{or} \quad m(k) = \lceil \log_\phi(\sqrt{k}) \rceil.$$

**Example 2.1.** For $k = 2^{10}$, $m(k) = 7$ and $t(k, 633) = N(k) = m(k) = 7$.

For $k = 2^{16}$, $m(k) = 12$ and $t(k, 40, 503) = N(k) = m(k) = 12$.

In the above examples, $N(k) = m(k)$. However, $N(k) < m(k)$ for some specific values of $k$; e.g. $k = 2^{12}$. (See Subsection 3.1, Case 1.)

# 3 Worst-Case Analysis of *JWA*

In this section, we show how to find the largest number of iterations $N(k)$ for every integer $k > 0$, and we exhibit all the values of $c$ corresponding to the worst case of *JWA*.

For $p \leq m = m(k)$ and $c > 0$ integer, let $I_p(k)$ and $J_p(k)$ be two sets defined as follows,

$$\begin{aligned} I_p(k) \ &= \ \{c \mid (F_p/F_{p+1})k < c < (F_{p+1}/F_{p+2})k\} \quad \text{for } \ p \text{ even}, \\ I_p(k) \ &= \ \{c \mid (F_{p+1}/F_{p+2})k < c < (F_p/F_{p+1})k\} \quad \text{for } \ p \text{ odd} \end{aligned}$$

and

$$J_p(k) \ = \ I_p(k) \cap; \{c \mid \ \gcd(k,c) = 1\}.$$

**Proposition 3.1.** *Let $k > 9$ (i.e. $m(k) \geq 3$), and let $c$ and $n$ be two positive integers such that* $\gcd(k,c) = 1$ *and $s \leq m(k) = m$. The four following properties hold*

*(i)* $c \in I_n(k) \implies k/c = [1, 1, \ldots, 1, x]$*, where* $[1, 1, \ldots, 1, x]$ *denotes a continued fraction having at least $n$ times a "1" (including the leftmost* 1*), and $x$ is a sequence of positive integers (see e.g. [1]).*

*(ii) If $J_{m-1}(k) \neq \emptyset$, then $N(k) = m$ or $m-1$.*

*(iii) If $J_{m-2}(k) \neq \emptyset$, then $N(k) = m$, $(m-1)$ or $(m-2)$.*

*(iv) If $k = 2^s$, $N(k) = m$, $(m-1)$ or $(m-2)$.*

*Proof.*
*(i)* Let $a_n/b_n = [1, 1, \ldots, 1] = F_{n+1}/F_n$ be the $n$-th convergent of the golden ratio $\phi$, containing $n$ times the value "1" (see [1, 4] for more details). To prove *(i)*, we show that $F_{n+1}/F_n$ is the $n$-th convergent of the rational number $k/c$; in other words,

$$|(k/c) - (F_{n+1}/F_n)| \; < \; 1/(F_n)^2. \tag{4}$$

Now, $(F_{n+1})^2 - F_n F_{n+2} = (-1)^n$ and, since $c \in I_n(k)$,

$$|(k/c) - (F_{n+1}/F_n)| \; < \; |(F_{n+1})^2 - F_n F_{n+2}|/(F_n F_{n+1}) \; = \; 1/(F_n F_{n+1}) \; < \; 1/(F_n)^2.$$

*(ii)* First, recall an invariant loop property which is also an Extended Euclidean Algorithm property: for $i = 1, \ldots, (t-1)$, where $t = t(k, c)$, we have that

$$n_i \, |d_{i+1}| \; + \; n_{i+1} \, |d_i| \; = \; k. \tag{5}$$

We first prove that $n_{m-2} > \sqrt{k}$. In fact, if we assume that $J_{m-1}(k) \neq \emptyset$, then from *(i)*, there exists an integer $c$ such that $k/c = [1, 1, \ldots, 1, x]$ with $(m-1)$ such 1's. Then, $q_i = 1$ and $|d_i| = F_{i+1}$ for $i = 1, \ldots, (m-1)$.
Now if $n_{m-2} < \sqrt{k}$, then, since $n_{m-1} < n_{m-2}$,

$$\begin{aligned} k \;&=\; n_{m-2} \, |d_{m-1}| \; + \; n_{m-1} \, |d_{m-2}| \; = \; n_{m-2} \, F_m \; + \; n_{m-1} \, F_{m-1} \\ &< \; \sqrt{k} \, (F_m + F_{m-1}) = \sqrt{k} \, F_{m+1}, \end{aligned}$$

and hence, $\sqrt{k} < F_{m+1}$, which contradicts the definition of $m(k)$, and $n_{m-2} > \sqrt{k}$. If $n_{m-1} < \sqrt{k}$, then $t(k, c) = m - 1$ and $N(k) \geq m - 1$; else, if $n_{m-1} > \sqrt{k}$, then $N(k) = m$.

*(iii)* The proof is similar to the previous one. There exists an integer $c$ such that $q_i = 1$ and $|di| = F_{i+1}$ for $i = 1, \ldots, (m-2)$. So, $n_{m-3} > \sqrt{k}$, and the result follows.

*(iv)* Let $\Delta_{m-2}$ be the size of the interval $I_{m-2}$. Then,

$$\begin{aligned} \Delta_{m-2} \;&=\; |(F_{m-2}/F_{m-1}) \, k \; - \; (F_{m-1}/F_m) \, k| \\ &=\; k \, |F_{m-2} F_m - (F_{m-1})^2| / (F_{m-1} F_m) \; = \; k/(F_{m-1} F_m). \end{aligned}$$

Since

$$2F_{m-1} F_m \; < \; (F_{m-1} + F_m)^2 = (F_{m+1})^2 \;\text{ and }\; (F_{m+1})^2 \leq k, \;\text{ then }\; \Delta_{m-2} > 2.$$

Thus, out of two consecutive values within $I_{m-2}(k)$, at least one integer is odd. Therefore, $J_{m-2}(k) \neq \emptyset$ and we can apply *(iii)*. (Note that this argument is not valid when $k$ is not a power of 2.) $\square$

**Remark 3.1.**

1. If $J_m(k) \neq \emptyset$, then $N(k) \geq m-1$, since $J_m(k) \subset J_{m-1}(k) \subset J_{m-2}(k)$.

2. The relation $N(k) = m-2$ holds for several $k$'s (e.g. for $k = 90$).

3. For any given integer $k$, there may exists a positive integer $c$ such that $c \notin J_m(k)$, whereas $t(k,c) = m$. Such is the case when $k = 15,849$: $m = 10$, $I_m(k) = \{9,795\}$ and, since $\gcd(k, 9, 795) \geq 3$, $J_m(k) = \emptyset$. However, for $c = 11,468$, $t(k, 11, 468) = 10$.

This last example shows that $J_m(k)$ is not made of all integers $c$ such that $t(k,c) = m$, with $\gcd(k,c) = 1$. Proposition 3.2 shows how to find all such numbers. For the purpose, two technical lemmas are needed first.

**Lemma 3.1.** *For every $m \geq 3$, the following three implications hold.*

*(i)* $\exists i \mid q_i = 2 \implies F_{m+1} + F_{m-1} \leq |d_m|$.

*(ii)* $\exists i \mid q_i \geq 3 \implies |d_m| \geq F_{m+2} > \sqrt{k}$.

*(iii)* $\exists i, j \ (i \neq j) \mid q_i = q_j = 2 \implies |d_m| \geq F_{m+2} + 2F_{m-3} > \sqrt{k}$.

*Proof.*
*(i)* Let $\Delta = \Delta(Q) = (\delta_i)_i$ be the sequence defined as: $\delta_{-1} = 0$, $\delta_0 = 1$, and $\delta_i = \delta_{i-2} + q_i\delta_{i-1}$, for $i = 1, 2, \ldots, m$ with $Q = (1, 2, 1, \ldots, 1)$. An easy calculation yields $\delta_i = F_{i+1} + F_{i-1}$ for $i = 1, 2, \ldots, m$.
On the other hand, let $(d_i)_i$ be a sequence satisfying (3). We show that $|d_m| \geq \delta_m = F_{m+1} + F_{m-1}$ $(m \geq 3)$, and $\Delta$ is thus leading to the smallest possible $|d_m|$ satisfying the assumption in *(i)*, i.e. $|d_m| = F_{m+1} + F_{m-1}$ $(m \geq 3)$. More precisely,

If $D = D(Q)$ with $Q = (2, 1, 1, \ldots, 1)$, then $|d_2| = 3$, $|d_3| = 5$, and $|d_m| = F_{m+2}$, whereas $\delta_2 = 3$, $\delta_3 = 4$ and $\delta_m = F_{m+1} + F_{m-1}$. Thus, $|d_m| > \delta_m$.

If $D = D(Q)$ with $Q = (1, 1, \ldots, 2, \ldots, 1)$ and $q_p = 2$ for some $p \geq 3$, then $|d_p| = F_{p-1} + 2F_p = F_{p+2}$ and $|d_{p+1}| = F_p + F_{p+2}$, whereas $\delta_p = F_{p+1} + F_{p-1}$ and $\delta_{p+1} = F_{p+2} + F_p$. It is then clear that $|d_i| > \delta_i$ for $i \geq p$, and $|d_m| \geq \delta_m = F_{m+1} + F_{m-1}$.

*(ii)* Similarly, let $\Delta = \Delta(Q)$ defined by $Q = (1, 3, 1, \ldots, 1)$, and let $D$ be a sequence satisfying the assumption. Then $|d_m| \geq \delta_m = F_{m+2}$ $(m \geq 3)$.

If $D = D(Q)$ with $Q = (3, 1, \ldots, 1)$, then $|d_2| = 4$, $|d_3| = 7$, whereas $\delta_2 = 4$ and $\delta_3 = 5$. Clearly, $|d_i| > \delta_i$ for $i = 3$, and $|d_m| > \delta_m > F_{m+2}$.

If $D = D(Q)$ with $Q = (1, 1, \ldots, 3, \ldots, 1)$ and $q_p = 3$ for $p = 3$, then $|d_p| = F_{p-1} + 3F_p = F_{p+3} + F_{p-2}$, and $|d_{p+1}| = F_{p+3} + F_p + F_{p-2}$, whereas $\delta_p = F_{p+2} + F_{p-3}$ and $\delta_{p+1} = F_{p+3} + F_{p-2}$. Therefore, $|d_i| \geq \delta_i$ for $i \geq p$, and $|d_m| \geq \delta_m = F_{m+2} + F_{m-3} > F_{m+2}$.

*(iii)* The proof is similar to the previous one with $Q = (1, 2, 1, \ldots, 1, 2, 1)$. For such a choice of $Q$, $|d_m| \geq \delta_m = F_{m+2} + 2F_{m-3}$, and the result follows. □

**Lemma 3.2.** *For every $m \geq 3$, let $Q = (1, 1, \ldots, 1, 2, 1, \ldots, 1)$, and let $p$ be the index such that $q_p = 2$ ($q_j = 1$ for $j \neq p$, $1 \leq j \leq m$). Then, for $p = 1, 2, \ldots, m$, $|d_m|$ explicitly expresses as*

$$|d_m| \;=\; F_{m-p+1}\, F_{p+2} \;+\; F_{m-p}\, F_p.$$

*Proof.* The proof proceeds along the same arguments as for Lemma 3.1. □

**Proposition 3.2.** *For every integer $k \geq 9$ ($m \geq 3$), if $t(k, c) = m$, then*

*either $c \in J_m(k)$,*

*or $k/c = [1, \ldots, 1, 2, 1, \ldots, 1, x]$. That is, there exists $i \in \{1, \ldots, m\}$ such that $q_i = 2$ and for any $j \neq i$, $j \leq m$ and $q_j = 1$.*

*In that case, the inequality $F_{m+1} + F_{m-1} < \sqrt{k}$ holds.*

*Proof.* The proof follows from the inequalities (2) and Lemma 3.1. □

## 3.1 Application of Proposition 3.2

The two following cases are examplified in Table 1. Assume $J_m(k) = \emptyset$.

**Case 1:** $N(k) \leq m(k) - 1$ holds, for example when $k = 2^6$, $2^8$ or $2^{12}$, etc. (the inequality $F_{m+1} + F_{m-1} > \sqrt{k}$ holds).

**Case 2:** $N(k) = m(k)$. The procedure that determines all possible integers $c$ in the worst case is described in Section 4.

## 4 Worst-Case Occurrences

Assuming that $J_m(k) = \emptyset$, we search for the positive integers $c$ such that $t(k,c) = m(k)$.

**Step 1.** Consider each value of $p$ ($p = 1, 2, \ldots, m$), and select the $p$'s that meet the condition $|d_m| < \sqrt{k}$ (Lemma 3.1 provides all values of $|d_m|$ for each such $m$). If $t(k,c)$ is still equal to $m$, then there exists a pair $(n_{m-1}, n_m)$ satisfying the Diophantine equation

$$n_{m-1}\,|d_m| \;+\; n_m\,|d_{m-1}| \;=\; k, \tag{6}$$

under the two conditions

$$\gcd(n_m, n_{m-1}) = 1 \quad \text{with} \qquad n_m < \sqrt{k} < n_{m-1} \tag{7}$$

and

$$0 \;<\; n_m,\, |d_m| \;<\; \sqrt{k}. \tag{8}$$

The system of equations (6)-(7)-(8) is denoted by $(\Sigma_Q)$, since it depends on $|d_m|$ and $|d_{m-1}|$, and thus on $Q$. Further, Eq. (6) is the expression of (5) when $i = m-1$, Eq. (8) expresses the exit test condition of *JWA* and Eq. (7) ensures that $\gcd(k,c) = \gcd(n_m, n_{m-1}) = 1$.

**Step 2.** Eq. (6) is solved modulo $|d_{m-1}|$. For $0 \le a < |d_{m-1}|$,

$$n_{m-1} \;\equiv\; k/|d_m| \pmod{|d_{m-1}|} \;\equiv\; a \pmod{|d_{m-1}|},$$

and, from the inequality

$$\sqrt{k} < n_{m-1} < k/|d_m|,$$

we have $n_{m-1} = a + r\,|d_{m-1}|$, where $r$ is a positive integer such that

$$(\sqrt{k} - a)/|d_{m-1}| \;<\; r \;<\; (k/|d_m| - a)/|d_{m-1}|.$$

Therefore, there exists only a finite number of solutions for $n_{m-1}$. Each solution of Eq. (6) (if any) fixes a positive integer $c \equiv n_{m-1}/|d_{m-1}| \pmod{k}$ such that $t(k,c) = m$, and $N(k) = m$.

**Example 4.1.** Let $k = 15,849$ and $m = 10$. By Lemma 3.2 (with $m = 10$ and $p = 2$), Eq. (6) yields $123n_{m-1} + 76n_m = 15,849$. Solving modulo 76 gives $n_{m-1} = 127$ and $n_m = 3$. The pair $(n_{m-1}, n_m)$ corresponds to the value $c = 11,468$, and $t(k,c) = N(k) = m(k) = 10$, while $J_m = \emptyset$.

### 4.1 Applications

The following algorithm summarizes the results by computing the values of $N(k)$.

$t := m$ ;
**repeat**
**if** $\exists c \in J_t | n_{t-1} > \sqrt{k}$ **then** $N := t$
**else** /* $J_t = \emptyset$ or
no $c \in J_t$ satisfies $n_{t-1} > \sqrt{k}$ */
**if** $(F_{t+1} + F_{t-1} < \sqrt{k})$
**and** ($\exists c$ solution of $(\Sigma_Q)$)
**then** $N := t$ **else** $t := t - 1$ ;
**until** $N$ is found

**Remark 4.1.**

1. The algorithm terminates, since $N(k) \geq 1$ for every $k \geq 3$. Indeed, the first condition in the repeat loop always holds when $t = 1$, since $k - 1 \in J_1(k)$ $(k \geq 3)$.

2. In the algorithm, $(\Sigma_Q)$ corresponds to the system (6)-(7)-(8), where $t$ substitutes for $m$.

The case when $k > 1$ is an even power of 2 is of special importance, since it is related to the practical implementation of *JWA* [8]. Table 1 gives some of the values of $N(k)$, for $k = 2^{2s}$ $(2 \leq s \leq 16)$.

| $k$ | $2^4$ | $2^6$ | $2^8$ | $2^{10}$ | $2^{12}$ | $2^{14}$ | $2^{16}$ | $2^{18}$ | $2^{20}$ | $2^{22}$ | $2^{24}$ | $2^{26}$ | $2^{28}$ | $2^{30}$ | $2^{32}$ |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| $m(k)$ | 3 | 5 | 6 | 7 | 9 | 10 | 12 | 13 | 15 | 16 | 17 | 19 | 20 | 22 | 23 |
| $N(k)$ | 2 | 4 | 5 | 7 | 8 | 10 | 12 | 12 | 14 | 15 | 16 | 19 | 20 | 21 | 22 |

Table 1: Values of $m(k)$ and $N(k)$ for $k = 2^{2s}$ $(2 \leq s \leq 16)$.

# 5 Concluding Remarks

First we must point out that the condition $\gcd(k, c) = 1$ is a very strong requirement: it eliminates many integers within $I_m(k)$ and many solutions of $(\Sigma_Q)$. This can be seen e.g. when $k = 2^{24}$. Then $m(k) = 17$, and the choice of $Q = (1, 2, 1, \ldots, 1)$, (i.e., $|d_m| = 3,571$, $|d_{m-1}| = 2,207$) yields $n_{m-1} = 4,404$ and $n_m = 476$, which leads to the solution $c = 12,140,108$. We still have $t(k, c) = m(k) = 17$ but unfortunately $\gcd(k, c) \neq 1$, and $N(k) = 16 = m(k) - 1$.

Checking whether $J_{m-2}(k)$ is empty is easy. It gives a straightforward answer to the question whether

$$m(k) - 2 \ \leq \ N(k) \ \leq \ m(k)$$

or not.

The following problems remain open.

- The example in Table 1 shows that, for $k = 2^{2s}$ $(2 \leq s \leq 16)$, the values of $N(k)$ are either $N(k) = m(k)$ or $N(k) = m(k) - 1$. Does the inequality

  $$m(k) - 1 \ \leq \ N(k)$$

  always hold for $k = 2^{2s}$ $(n \geq 2)$?

- $N(k)$ is never less than $m(k) - 2$. Are the inequalities

  $$m(k) - 2 \leq N(k) \leq m(k)$$

  true for every positive integer $k \geq 9$?

- Find the greatest lower bound of $N(k)$ as a function of $m(k)$.

# References


[1] G.H. Hardy and E.M. Wright, *An Introduction to the Theory of Numbers*, Oxford University Press, London, 1979.

[2] T. Jebelean, A Generalization of the Binary GCD Algorithm, *in Proc. of the Int. Symp. on Symbolic and Algebraic Computation (ISSAC'93)*, 1993, 111-116.

[3] T. Jebelean, An Algorithm for Exact Division, *J. of Symbolic Computation*, 15, 1993, 169-180.

[4] D.E. Knuth, *The Art of Computer Programming: seminumerical algorithms*, Vol. 2, 2nd ed., Addisson Wesley, 1981.

[5] M.S. Sedjelmaci, C. Lavault, Improvements on the Accelerated Integer GCD Algorithm, *Information Processing Letters*, 61, 1997, 31-36.

[6] M.S. Sedjelmaci and C. Lavault, A new modular division algorithm and applications, *in Proc. of the 6th Int. Conf. on Theoretical Computer Science (ICTCS'98)*, World Scientific, 1998, 65-76.

[7] J. Sorenson, Two Fast GCD Algorithms, *J. of Algorithms*, 16, 1994, 110-144.

[8] K. Weber, Parallel Implementation of the Accelerated Integer GCD Algorithm, *J. of Symbolic Computation*, 21, 1996, 457-466.

Christian LAVAULT* and Sidi Mohamed SEDJELMACI†
LIPN, CNRS UPRES-A 7030 – `http://lipn.univ-paris13.fr`
Université Paris 13, 99 av. J.-B. Clément F-93430 Villetaneuse.
* *E-mail:* `Christian.Lavault@lipn.univ-paris13.fr`,
*URL:* `http://lipn.univ-paris13.fr/~lavault`
† *E-mail:* `sms@lipn.univ-paris13.fr`